Received: 5 October 2021 | Accepted: 12 December 2022

DOI: 10.1111/2041-210X.14053

Methods in Ecology and Evolution

APPLICATION

starve: An R package for spatio-temporal analysis of research survey data using nearest-neighbour Gaussian processes

Ethan Lawler | **Chris Field** | **Joanna Mills Flemming**

Department of Mathematics and Statistics, Dalhousie University, Halifax, Nova Scotia, Canada

Correspondence
Ethan Lawler
Email: lawlerem@dal.ca

Funding information
Canadian Statistical Science Institute; Killam Predoctoral Scholarship; Ocean Frontier Institute; Vanier Canada Graduate Scholarship

Handling Editor: Saras Windecker

Abstract

1. Spatio-temporal datasets that are difficult to analyse are commonly derived from ecological surveys. There are software packages available to analyse these datasets, but many of them require advanced coding skills. There is a growing need for easy-to-use packages that researchers can use to analyse common ecological datasets.
2. We develop a particular generalized linear mixed model framework for spatio-temporal point-referenced data that is flexible enough to accommodate data from most ecological surveys while being structured enough to facilitate analyses without advanced coding. Our implementation in the **starve** package uses a computationally efficient version of a nearest-neighbour Gaussian process enabling analysis of relatively large datasets.
3. A tutorial analysis of a Carolina wren survey presents a recommended workflow for analyses while showcasing the capabilities of the package.
4. Our model and package are tools that can easily be added to researchers' routine to help make sense of data from ecological surveys. We emphasize the ability of our model to create fine-scale spatio-temporal predictions which can then be used to visualize and identify important trends in species distributions.

KEYWORDS
generalized linear mixed model, hierarchical model, nearest-neighbour Gaussian process, software, spatio-temporal analysis, species distribution model

1 | INTRODUCTION

Many questions in ecology are directly related to the spatial distribution of a species and to changes in their distribution and overall abundance over time. As a result many ecological data sources consist of scientific research surveys which record the count, weight or presence/absence of a species along small spatial transects within the study area. These surveys are typically repeated at regular intervals, commonly with transects being surveyed in the same month every year. Some surveys follow a repeated transect sampling design that sample the same spatial transects every year. Others use a stratified random sampling design which leads to different transects in the study area being sampled each year. Some concrete examples will highlight some of the characteristics of these datasets and some of their potential uses.

- **eBird**: A real-time, online checklist program, eBird has revolutionized the way that the birding community reports and accesses information about birds. The database provides rich data with basic information on bird abundance and distribution at a variety

of spatial and temporal scales. If we consider Nova Scotia, Canada there are about 25,000 checklists provided by about 600 citizen-science observers on a yearly basis with about 350 species reported annually. Given the huge amount of data collected in this database, tools that provide quick and easy to understand visualizations of the data would help ecologists make full use of efforts of the citizen scientists.

- **Disease Mapping**: Some epidemiology datasets have the same characteristics as ecological research surveys (Giorgi et al., 2018). However many disease prevalence datasets must comply with privacy regulations, and thus the data must either be analysed in-house or reported as summaries over larger areas. If the data are analysed in-house then the research team either needs access to easy-to-use to software to help analyse their datasets, or employ resident statisticians that have the expertise to analyse the data from scratch and also have the necessary clearance to actually view the data (which is not always easy to do!).
- **Fisheries Science**: The original motivating example for the **starve** package introduced here is the analysis of fisheries data in Atlantic Canada, with a biology Master's student using the **starve** package to lead the data analysis (Jubinville et al., 2021). In that research we used surveys following the geographically stratified random sampling design to create predictions of the spatial distribution for both at-risk skate species and commercially valuable target species such as cod and haddock. We then used these predictions to identify areas of the ocean where fishing vessels trying to catch the commercially valuable species had more risk of catching the at-risk skate species as a byproduct of the fishing process. If included as part of a fisheries management plan these results could help direct fishing effort to target areas with a relatively high abundance of the target species but with a low risk of bycatch. The privacy issues mentioned above for epidemiological datasets can also apply to fisheries science when using data collected from commercial fishing vessels.

Statistical models for these surveys typically fall under the name of 'species distribution models', where one of the main goals is to produce a predicted map of the spatial distribution of the species. Other goals for species distribution models include identifying important environmental factors that influence species abundance, or estimating the interaction between different species. See Robinson et al. (2017) for a review of species distribution modelling in the context of fisheries, and for a review of species distribution models in a broader ecological context including a discussion of the statistical concepts that underlie most of these models. Repeated transect designs are much easier to analyse because they can be formulated as multivariate time series models, and associated spatially explicit models applicable to this design have a relatively long history in the statistical literature (Eynon & Switzer, 1983; Goodall & Mardia, 1994). The geographically stratified random sampling designs have traditionally been modelled on aggregate by discarding the exact spatial coordinates but modelling each strata in its own independent analysis, creating a pseudo-spatial analysis. Models that keep the entirety of the spatial information have only started to appear recently in the ecology literature (Berger et al., 2017; Cosandey-Godin et al., 2014). In addition, recent applied interest in analysing large spatial or spatio-temporal datasets has led to theoretical innovations that make these models computationally tractable for wide use, with a comparative study for these innovations given in (Heaton et al., 2018).

Alongside the introduction of these statistical models has come the development of software packages, making the models readily available to applied researchers. Perhaps the most widely used package for spatial ecology is **R-INLA** (Lindgren & Rue, 2015) which implements the so called stochastic partial differential equation approach for Gaussian random fields (Lindgren et al., 2011). Another popular package in spatial ecology is Template Model Builder, or **TMB** (Kristensen et al., 2016), which takes advantage of the Laplace approximation and automatic differentiation for fast optimization of a user-coded likelihood function. Both **INLA** and **TMB** are quite flexible but require the user to spend a significant amount of time learning how to code the model or likelihood function.

In addition to those two general-purpose packages are a suite of more specialized **R** packages that are easy to use in specific domains. A few examples include **LatticeKrig** (Nychka et al., 2016) for using purely spatial data to create fine-scale predictions and **SpatioTemporal** (Lindström et al., 2013) for fitting Gaussian spatio-temporal processes using basis functions. Ecology-centric packages that support spatial analyses include **Hmsc** (Tikhonov et al., 2020) for estimating multi-species community data and the **VAST** package (Thorson & Barnett, 2017) which is tailored towards fisheries research and supports spatio-temporal data. What is missing from this suite is a package that (1) supports computationally efficient spatio-temporal analysis and predictions for the data types encountered in ecological surveys, in particular counts, weights and presence/absence data, (2) provides a simple interface to support wide use of the model and a streamlined workflow and (3) natively uses the spatial data formats provided by **R**'s rich spatial data ecosystem (see the CRAN Task View on Analysis of Spatial Data; Bivand & Nowosad, 2022). **VAST** fits most of these criteria, however, it does not natively use spatial data formats and the strong emphasis it places on fisheries-specific models may limit its attractiveness to researchers not in fisheries.

To fill this niche we have developed the **starve** package for the **S**patio-**T**emporal **A**nalysis of **R**esearch sur**VE**y data. It fits our three criteria by (1) using **TMB** to implement a new spatio-temporal hierarchical model that incorporates recent research into computationally efficient spatial statistical modelling, (2) performing the most common analysis tasks (model fitting, simulations and predicting) using only three main functions whose syntax resembles that of typical calls to *glm* and *predict* functions familiar from introductory **R** courses and (3) directly accepts spatial data from the **sf** package and outputs the spatial data formats provided by the **sf**, **raster** and **stars** packages. We first introduce the basic statistical model implemented by the package leaving most of the details to technical appendices. Then we work through an example analysis

of a Carolina wren survey that demonstrates the workflow of the **starve** package.

2 | MODELLING FRAMEWORK

The model developed for the **starve** package is a hierarchical model with levels of the hierarchy that partition the model into logical components based on which source of variability they account for. There are four levels in the hierarchy: the 'temporal' level models the global change of the species distribution from year to year, the 'spatio-temporal' level models the spatio-temporal variability in the species distribution on top of the global change provided by the 'temporal' level, the 'linear' level adds the effect of any covariates on top of the spatio-temporal distribution provided by the 'spatio-temporal' level, and the 'response' level converts the 'linear' level to the scale of the data and accounts for any leftover variability not accounted for otherwise.

The 'linear' and 'response' levels can be written in the standard generalized linear mixed model framework:

$$Y_i | \mu_i \sim f_\theta(y_i; \mu_i),$$

$$\mu_i | w = g^{-1}(X_i\beta + Z_i w),$$

where Y_i is the response variable, $f_\theta(y_i; \mu_i)$ is a response distribution with parameters θ and mean μ_i, g^{-1} is an inverse link function, X_i is the ith row of a design matrix for fixed effects/covariates with regression coefficients β, and Z_i is the ith row of a design matrix for spatio-temporal random effects w.

The innovation provided in the **starve** package is in modelling the random effects w, which we do in the 'temporal' and 'spatio-temporal' levels of the hierarchy. Most spatial and spatio-temporal models are based on Gaussian random fields. Ours is no different so we briefly describe them. A Gaussian random field is a generalization of a multivariate Gaussian distribution used to model, among other things, variables that change with location such as the presence or absence of a species throughout a geographic area. They are described by a mean function $\mu(s)$ that gives the expected spatial distribution of the variable, and a covariance function $C(s_1, s_2)$ that encodes Tobler's first law of geography where "everything is related to everything else, but near things are more related than distant things" (Tobler, 1970). An introduction to Gaussian process for time series, where they are easier to visualize and understand, is given by (Roberts et al., 2013).

The main problem with Gaussian processes is that working with them requires the inversion of large covariance matrices, which can be computationally prohibitive for the size of most modern datasets. Many changes and approximations to Gaussian processes to make them computationally feasible have been proposed in the literature (the stochastic partial differential equation approach used by **INLA** is one of them). The model implemented in the **starve** package uses a modification called the nearest-neighbour Gaussian process (Datta et al., 2016). The idea behind the nearest-neighbour Gaussian process is that 'everything is related to everything else, but near things are more related than distant things so it's probably OK to forget about the distant things so that the models run much faster'. A graphical description of how nearest-neighbour Gaussian processes work is given in Figure 1.

We further adapted the nearest-neighbour Gaussian process to use in the 'spatio-temporal' level of our model. We choose a set of locations for the random effects that we will model every year, which we call the persistent graph nodes (see Appendix S1). The persistent graph nodes help incorporate the global change from the 'temporal' level of the model into the 'spatio-temporal' level. These nodes do not need to be the same as the data locations though it usually makes sense to use all of or a subset of the data locations. If there are any data locations not used, then we model those locations only for the years that have observations at that location.

The 'temporal' level of the hierarchy represents a second set of random effects, one for each year, which are modelled as an AR(1) time series. The values of these random effects are used in the mean function for the 'spatio-temporal' level of the hierarchy to ensure that each individual location also follows an AR(1) time series with the same properties as the one in the 'temporal' level.

3 | AN EXAMPLE ANALYSIS

Included in the **starve** package is a relatively simple survey dataset containing the number of sightings of Carolina wren *Thryothorus ludovicianus* recorded once a year along spatial transects in Missouri, U.S.A. taken from the package **STRbook**, which in turn was modified from (Pardieck et al., 2017). We will use these data to produce a smooth map of the wren's spatial distribution during the years covered by the dataset, and forecast their spatial distribution a short time ahead into the future. This analysis is meant to demonstrate the use of the **starve** package by working through the often unreported typical steps—from model fitting and checking to predictions and interpretation. We also highlight some of the surrounding **R** packages that a typical analysis might use. The produced map could then be used to identify shifts in spatial distribution and relative abundance over time, and can be used to set expectations for how the distribution of the species could look like in the near future.

The Carolina wren dataset is stored as point data using the 'simple features' open standard for spatial vector data as implemented in the **R** package **sf** (Pebesma, 2018). This is a common and straightforward spatial data format that is supported by many **R** packages, packages in other languages, and in GIS programs. We could not find the coordinate reference system used for these data, so for demonstrative purposes we assume it is WGS84. The dataset has 783 observations across 21 years, ranging from 1994 to 2014. The same 68 locations are sampled every year, however, there are some missing transects which are not included in the dataset. Without these missing transects there is a median 36 observations per year, with as few as 26 and as many as 45 in a single year. The data are mapped in Figure 4.

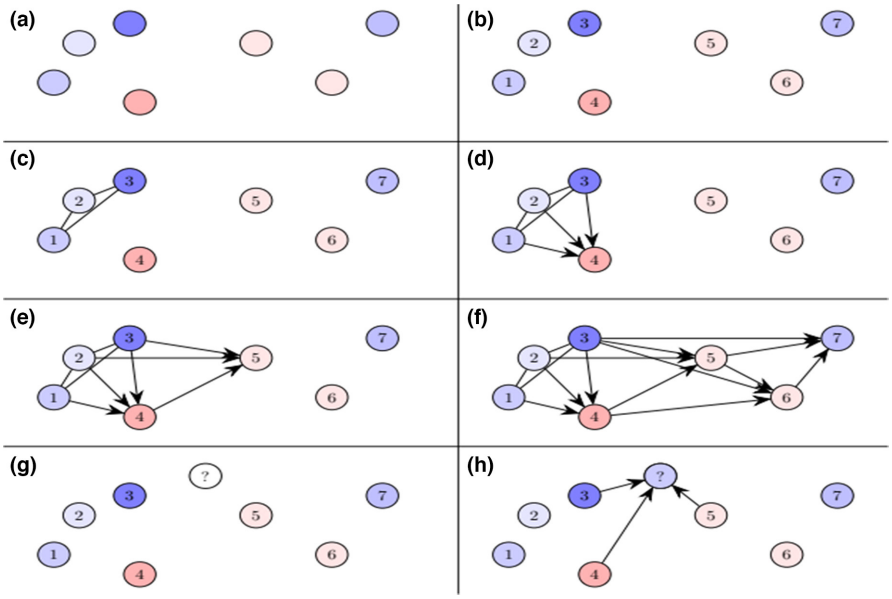

FIGURE 1 Graphical model representation of the steps for computing the likelihood of a nearest-neighbour Gaussian process using $k = 3$ nearest neighbours. (a) There are seven observed locations, with the value of each observation given by the colour of each circle. (b) Order the observed locations. Observations that are close in space should be close together in the ordering. In this example we put the locations in increasing order from left to right. (c) Find the joint multivariate normal distribution for the first $k = 3$ locations implied by the mean function and covariance function. (d) Compute the conditional distribution of the fourth location given its $k = 3$ nearest neighbours that came before it in the ordering, in this case locations 1, 2 and 3. (e) Compute the conditional distribution of the fifth location given its $k = 3$ nearest neighbours that came before it in the ordering, in this case locations 2, 3 and 4. (f) Continue with the conditional distributions for each of the remaining locations: in this example the conditional distribution of location 6 given locations 3, 4 and 5; and the conditional distribution of location 7 given locations 3, 5 and 6. (g) We want to predict the value of the spatial field at a new unobserved location. (h) The predictive distribution of the new location is the conditional distribution of that location given the $k = 3$ nearest observed locations, regardless of where those locations fall in the ordering, in this case locations 3, 4 and 5.

3.1 | Creating and fitting a model object with the strv_prepare function

Before fitting a model there are a few data preprocessing steps that are run, such as creating the graph used in the nearest-neighbour Gaussian process and creating a model object that holds the model parameters and spatio-temporal random effects. These preprocessing steps are all performed automatically as part of the **strv_prepare** function. The very first step in analysing the wren dataset is to use this function to create the model object describing the model formula, the dataset we want to analyse, and the response distribution for the data. Our model formula needs to specify that the survey count is the response variable and that the year of the survey is the time index. The spatial coordinates for each observation point are automatically detected from the supplied data. Since we are analysing counts we will use a Poisson distribution with a log link function for our initial model.

```
set.seed(30795)
bird_fit<- strv_prepare(
  cnt ~ time(year),
  bird_survey,
  distribution ="poisson",
  fit =TRUE
)
```

The first argument to the **strv_prepare** function is the model formula. The variable on the left-hand side gives the column name of the response variable. The right-hand side of the formula can tell the model which covariates, if any, to use in the analysis. While our dataset does not have any covariates we can use, if we wanted to include the linear effect of an elevation covariate and a temperature covariate we could use familiar **R** formula syntax, for example, **cnt ~ elev + temp + time(year)**. The right-hand side is also used to tell the model which variable should be used for the time index by enclosing the name of that variable in the **time(...)** special function. The right-hand side of the formula can be used to specify other model components such as sample sizes for binomial or tweedie response distributions, for setting the spatial covariance function, or for using more complicated forms for covariate effects. The full range of use for the model formula is described in the **R** documentation for the **strv_prepare** function.

The second argument is the dataset we want to use. This dataset must be a **sf** data.frame with point geometries, and must have any variables named in the model formula. The spatial information is automatically detected and used from the geometry column of the data.frame, so it does not need to be specified in the model formula.

The **distribution** argument tells the model which distribution should be used for the response variable, and chooses a default link

function depending on the response distribution. If a link function besides the default is wanted, you can override the default by using the **link** argument to the function. For our analysis we will use the default log link function.

The **strv_prepare** function also accepts arguments used to control how the nearest-neighbour graph is constructed. By default the persistent graph nodes are taken to be the unique locations present in the dataset, but a different set of node locations can be used by supplying a second **sf** data.frame with point geometries to the **nodes** argument. The main reason to use a different set of node locations is to use a smaller set of locations in order to decrease the computation time needed to fit the model, but this comes at the cost of using coarser approximations in describing how the spatio-temporal random effects evolve from one time to the next. You can also set the number of neighbours used by giving an integer to the **n_neighbours** argument, which defaults to 10. Smaller values for the number of neighbours will speed up computation time at the cost of using a more aggressive approximation, but the decrease in accuracy may not be significant (see Figure 2; Table 2).

The **fit = TRUE** argument tells the package that we want to fit the model after creating the model object. If this is the case then the starting values for the parameter estimates will be set to some default values. Sometimes the starting values will need to be chosen carefully in which case you can use **fit = FALSE**, modify the parameter values in the resulting model object to change the starting values, and then call the **strv_fit** function to fit the model. Details on how to do this are given in the package documentation.

The output of the **strv_prepare** function is a model object with all of the preprocessing steps done, and if **fit = TRUE** the parameter and random effect values held in the model object will be those estimated via maximum likelihood. The model object looks like this:

```
bird_fit
## A starve model object
##
## Model formula: cnt ~ time(year)
## Response distribution: poisson
## Link function: log
```

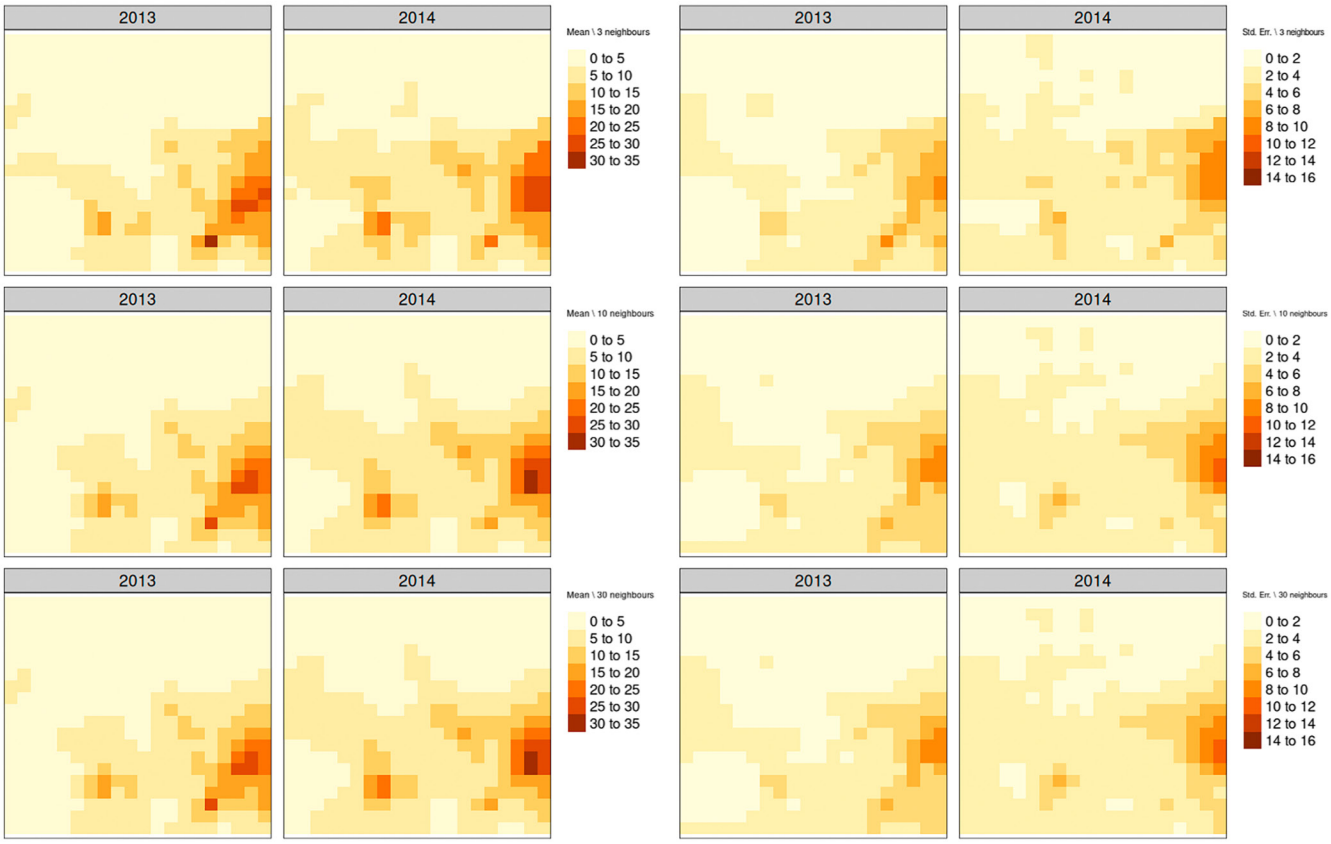

FIGURE 2 Predicted response mean (left) and standard error (right) using different numbers of nearest neighbours—$n = 3$ (top), $n = 10$ (middle), $n = 30$ (bottom)—in a **starve** model. The predictions are made for a 20×20 grid in the spatial extent of the Carolina wren data in the years 2013 and 2014. The predicted means and standard errors when using $n = 10$ are essentially identical to the predictions when $n = 30$, with any difference in the predictions small enough that the rasters are identical after applying the colour breaks. The predicted means and standard errors when using $n = 3$ are slightly different from either $n = 10$ or $n = 30$, although the differences are small enough that at quick glance they look the same. This suggests that it is feasible to use a small number of neighbours to drastically decrease computation time and memory usage (Table 2) while not losing much predictive accuracy.

```
## Optimizer message: relative convergence (4)
##
## Data is a simple feature collection with 783 features
## CRS: WGS 84
## Bounding times: 1994 2014
## Bounding box:
##       xmin       ymin       xmax       ymax
## -95.23445   36.11935  -89.24779   40.47128
```

Printing the object gives a brief description of the model including the model formula, response distribution, link function, optimizer message and a summary of the data. The convergence message is a diagnostic message from the optimizer used to find the maximum likelihood estimates of the parameters and random effects. Here the message is 'relative convergence', so we are confident that we have found a local maximum of the log-likelihood function. If the message does not indicate 'relative convergence' then a few strategies can be tried such as choosing different starting values or holding the spatial range parameter fixed at a somewhat large value. Diagnosing problematic convergence messages is in general a difficult task so some experimentation may be necessary to find the cause of such problems.

The parameter estimates for the different model components can be viewed through accessor functions such as **time_parameters** and **space_parameters**, for example:

```
space_parameters(bird_fit)
##    $cnt
##                      par          se  fixed
##    sd          0.05273397 0.005281752  FALSE
##    range      60.48205683 15.760156240  FALSE
##    nu          0.50000000 0.000000000   TRUE
```

The **par** column gives the parameter estimate for that parameter, the **se** column gives the standard error for the parameter estimate, and the **fixed** column determines if that parameter was held fixed at the given parameter value when fitting the model. After checking the optimizer message for successful convergence and inspecting the model parameters, the model fit should be checked to see if there is any leftover variation or structure in the data that the model did not account for.

3.2 | Checking the model fit with the strv_simulate function

With ecological count data it is important to check for over- or under-dispersion of the data relative to the fitted model. While over- or under-dispersion can be caused by any number of model mis-specifications, the easiest one to check is for mis-specification of the response distribution. We use the **strv_simulate** function in conjunction with the **DHARMa** package, which uses simulations to construct a parametric bootstrap estimator of the cumulative distribution function (CDF) for each data point (Hartig, 2020). The bootstrapped CDFs can then be used to calculate quantile residuals for the data, such that if the model is correct then the residuals will be uniformly distributed on the interval [0,1].

We will compare the wren dataset to one simulated set of observations each from a model with a Poisson response distribution, an over-dispersed negative binomial response distribution, and an under-dispersed Conway–Maxwell–Poisson response distribution. We then simulate 100 sets of observations from the fitted Poisson model to construct the bootstrap CDF. Quantile residuals for the Carolina wren dataset and each of the different response distributions are computed using this bootstrap CDF. Through this procedure we have four sets of residuals: one set where we know the model is correct (Poisson simulation), one set where we know the data are over-dispersed relative to the model (negative binomial simulation), one set where we know the data are under-dispersed relative to the model (Conway–Maxwell–Poisson simulation), and the final set of residuals coming from the real dataset. We then compare the residuals from the Carolina wren dataset to the residuals coming from the simulated datasets to determine if the data exhibit under- or over-dispersion relative to the fitted Poisson model. To ensure we are checking only the response distribution we simulate new observations conditional on the fitted random effects and parameters, so that all of the simulated datasets share the same spatio-temporal pattern as the Carolina wren dataset.

The Conway–Maxwell–Poisson distribution can be seen as a generalized version of the Poisson distribution that can exhibit both under- and over-dispersion and has been used in a variety of applications (Sellers et al., 2012). The Conway–Maxwell–Poisson distribution is under-dispersed when the dispersion parameter is less than 1, over-dispersed when greater than 1 and becomes a Poisson distribution when equal to 1.

We will only show the code to simulate the Poisson realization, since the code for simulating from the other realizations is identical after changing the response distribution to their respective values.

```
pois_sim<- strv_simulate(
  bird_fit,
  conditional = TRUE
)
pois_sim
## A starve model object
##
## Model formula: cnt ~ time(year)
## Response distribution: poisson
## Link function: log
## Optimizer message: Simulated realization from model
##
## Data is a simple feature collection with 783 features
## CRS: WGS 84
## Bounding times: 1994 2014
## Bounding box:
##       xmin       ymin       xmax       ymax
## -95.23445   36.11935  -89.24779   40.47128
```

The first argument to **strv_simulate** is a starve model object. A new set of random effects and data will be simulated using the parameter values held in the model object, whether defined by the user or the estimates from a fitted model. We want to use the same spatio-temporal random effects for each of our simulations, so we use the argument **conditional = TRUE** to keep the random effects held in the given model object and only simulate a new dataset conditional on those random effects. The standard errors of the parameter values are also set to NA to indicate that the parameter values are not estimates from data.

```
# Simulate 100 new datasets and combine the new simulated observations
# into a matrix where each column is a different simulation.
#
# These will be used to create the bootstrap CDF
pois_CDF<-do.call(
  cbind,
  mclapply(
    seq(100),
    function(i) {
      sim<-strv_simulate(
        bird_fit,
        conditional =TRUE
      )
      return(dat(sim)$cnt)
    },
    mc.cores =8
  )
)
# Compute the quantile residuals for the wren
# dataset relative to the fitted Poisson model
library(DHARMa)
bird_dharma<- createDHARMa(
simulated = pois_CDF,
observed = dat(bird_fit)$cnt,
integer = TRUE
)
```

The residual plot (Figure 3) suggests that the Carolina wren data exhibit under-dispersion relative to the Poisson model. To test the magnitude of the effect we refit the data using a Conway–Maxwell–Poisson distribution to more formally check if the under-dispersion suggested by the residual analysis is significant. To avoid repeating the preprocessing steps we create a copy of our original fitted model, change the response distribution to the Conway–Maxwell–Poisson distribution, then run the **strv_fit** function to estimate the parameters.

```
# Copy the existing model
bird_compois<-bird_fit
# Set the new response distribution
response_distribution(bird_compois)<-"compois"
# Fit the model. `silent = TRUE' suppresses the optimizer tracing
bird_compois<-strv_fit(
bird_compois,
silent =TRUE
)
# Print the convergence message
convergence(bird_compois)
## [1] "relative convergence (4)"
space_parameters(bird_compois)
## $cnt
##                par          se    fixed
## sd       0.0456067 0.006577758    FALSE
## range   70.0321382 22.241994976  FALSE
## nu       0.5000000 0.000000000    TRUE
# Print the estimated parameters for the
# Conway-Maxwell-Poisson distribution
response_parameters(bird_compois)
## $cnt
##                par          se    fixed
## dispersion 1.222485 0.1594049    FALSE
```

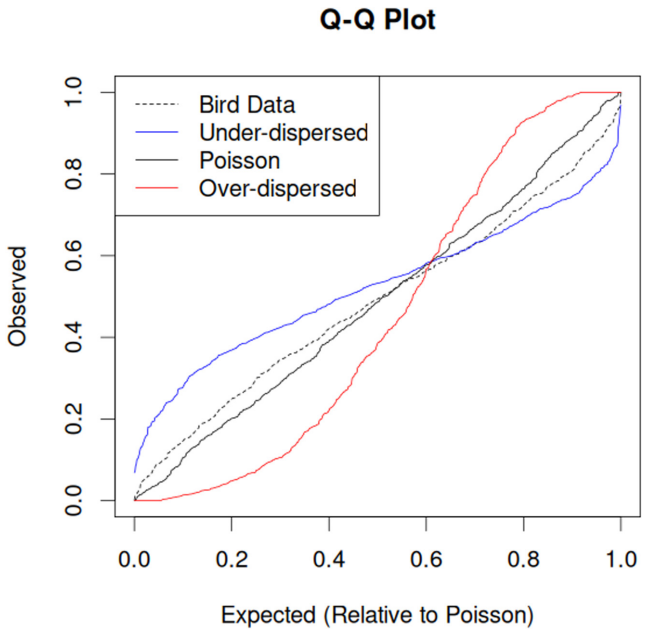

FIGURE 3 Q–Q plot comparing the residual patterns of the Carolina wren dataset to the residual patterns of simulated datasets with a known Poisson distribution or known under- or over-dispersion relative to a Poisson distribution. The under-dispersed residuals come a data simulated from a Conway–Maxwell–Poisson distribution with dispersion parameter equal to 0.2, the over-dispersed residuals from a negative binomial distribution with over-dispersion parameter equal to 4. The simulated datasets share the same spatio-temporal random effects as the Carolina wren dataset to isolate the effects of the response distribution. The Carolina wren residuals resembles the shape the under-dispersed Conway–Maxwell–Poisson residuals with both starting above the line for the Poisson residuals, crossing the Poisson line when the expected is roughly 0.6, and then ending below the line for the Poisson residuals. However, the amount of under-dispersion in the wren dataset is not as extreme as in the simulated data since the residuals are in between those of the Poisson simulations and the Conway–Maxwell–Poisson simulations.

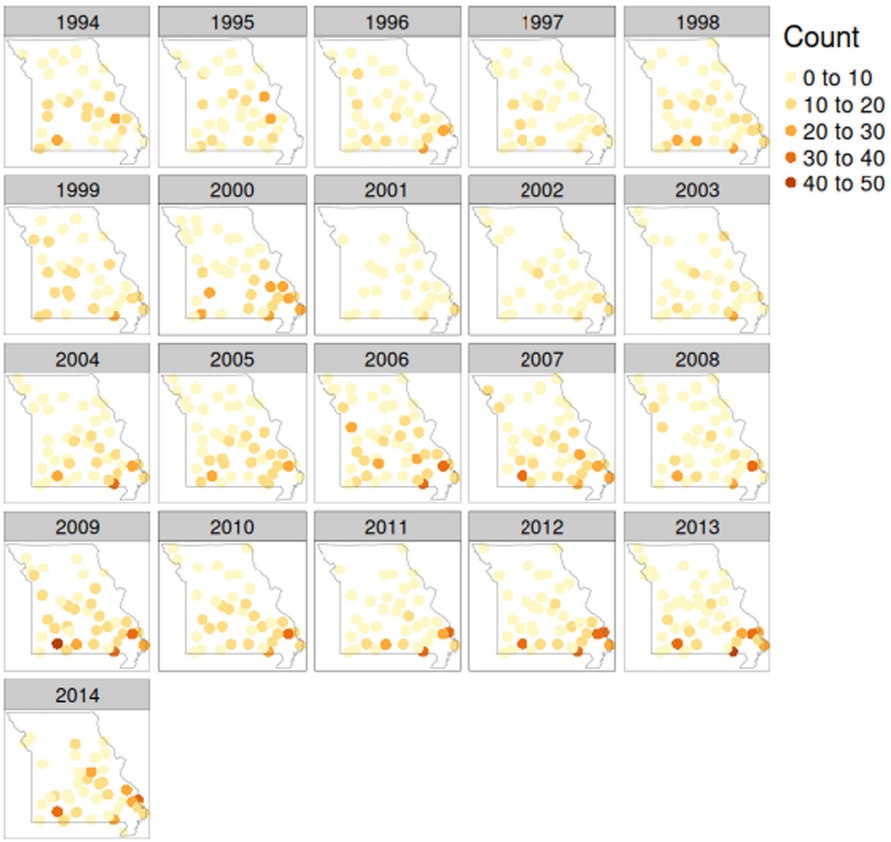

FIGURE 4 Map of observed counts for the Carolina wren dataset collected in Missouri. Each point represents the number of observed birds counted in a survey transect at that location. Some general trends can be gleaned from this map notably that most of the population resides in the southern part of the state, and the large and sudden decline of the population in 2001 followed by a slow recovery. More fine-scale trends are harder to identify from the point data, even when the same transect locations are used every year like they are here.

The Conway–Maxwell–Poisson model also successfully converged. While the estimated dispersion parameter surprisingly is in the range indicating over-dispersion, the standard error is large enough that the approximate 95% confidence interval = (0.910, 1.535) contains values covering under-dispersion, over-dispersion and the Poisson distribution. We attribute the discrepancy between the Q–Q plot and the parameter estimate to the combination of two things: first that if there is any under- or over-dispersion of the data relative to the Poisson model then the magnitude of it is too small to be estimated from the data, and second that the change in response distribution brought with it a change in estimates of the spatial parameters and spatio-temporal random effects. More generally, this speaks to the difficulty of model validation for spatio-temporal generalized linear mixed models and that more research needs to be done to understand it.

We will continue our analysis using the original Poisson model, although in practice we might perform more model validation steps or proceed with both models and check that the resulting predictions agree. The complete table of estimated model parameters for the Poisson model is given in Table 1.

TABLE 1 Parameter estimates and standard errors for Poisson model fitted to the Carolina wren dataset. Direct interpretation of the temporal and spatial parameter estimates may not be very meaningful. A better way to interpret the fitted model is to interpret the random effect and response mean predictions directly, for example using prediction maps like Figures 5 and 6. If the model had included any covariate effects or if there were parameters in the response distribution, then those parameter estimates could be interpreted directly

	Estimate	Standard error
(log) global mean μ	1.772	0.493
AR(1) correlation ϕ	0.905	0.020
temporal std. dev. σ	0.283	0.059
spatial std. dev. τ	0.053	0.005
spatial range ρ	60.482	15.760

3.3 | Predicting at unobserved locations/times with the strv_predict function

The final step in our analysis is to use the fitted model to produce a map of fine-scale predictions at unobserved locations and to forecast into the immediate future. We use the **strv_predict** function with the fitted Poisson model to predict the mean intensity, which should be roughly proportional to species abundance, throughout the state of Missouri in the years spanned by the dataset (1994–2014) and forecasts up to 4 years ahead (2015–2018). We want a smooth map so we first create a template raster to tell where predictions should be made. We chose the resolution of the raster based on the rough spatial resolution of the data, but a lower resolution raster could be used to decrease computation time at the cost of possibly smoothing over any small-scale spatial variability in the data. To save a bit of computation time we mask the template raster so that values of the raster are NA outside of Missouri state lines, which tells the **strv_predict** function to skip these raster cells. The boundary of Missouri was obtained from the **rnaturalearth** package (South, 2017).

```
missouri<- subset(
  ne_states(iso_a2 ="US", returnclass ="sf"),
    name =="Missouri",
    select ="name"
)
st_crs(missouri)<- 4326
raster_to_pred<- rasterize(
  missouri,
  raster(missouri, nrow =20, ncol =20),
  getCover =TRUE
)
raster_to_pred[raster_to_pred ==0]<- NA

bird_stars<- strv_predict(
  bird_fit,
  raster_to_pred,
  time =1994:2018
)
bird_stars
## stars object with 4 dimensions and 6 attributes
## attribute(s):
##                  Min.     1st Qu.    Median      Mean   3rd Qu.     Max.  NAs
## w          -0.3844453   1.1268977  1.672641 1.6203511 2.1046658 3.412355 3275
## w_se        0.2012790   0.3486235  0.404807 0.4519355 0.5079299 0.998019 3275
## linear     -0.3844453   1.1268977  1.672641 1.6203511 2.1046658 3.412355 3275
## linear_se   0.2012790   0.3486235  0.404807 0.4519355 0.5079299 0.998019 3275
## response    0.6808282   3.0860679  5.326215 6.1858578 8.2043600 30.336599 3275
## response_se 0.3595204   1.4481210  2.304867 2.7891404 3.5704265 15.233438 3275
## dimension(s):
##          from  to    offset     delta  ref  sys  point          values x/y
## x           1  20   -95.773  0.332708  WGS   84  FALSE            NULL [x]
## y           1  20   40.6188 -0.231631  WGS   84  FALSE            NULL [y]
## year        1  25      1994         1   NA   NA      1994,...,2018
## variable    1   1        NA        NA   NA   NA                cnt
```

The **strv_predict** function then takes the fitted model, the template raster and the vector of prediction years and outputs the predictions produced at the midpoint of every raster cell every year. Predictions can also be made for specific locations and times by supplying a **sf** data.frame containing a row for each specific (location, time) point. The output of **strv_predict** is a more general version of a raster called a data cube which allows the inclusion of a time coordinate, and is implemented as a **stars** object in the **stars** package (Pebesma, 2022). The predictions include a layer for the point predictions and standard error of the spatio-temporal random effects (**w** and **w_se**), the response mean after including any covariates but before applying the link function (**linear** and **linear_se**), and the response mean on the scale of the observations after applying the link function (**response** and **response_se**). If there are any covariates used to fit the model then they will also need to be given to the **strv_predict** function.

We give the predictions for the response mean predictions and standard errors directly to the **tmap** package (Tennekes, 2018), a general-purpose mapping package that can use the same **sf** data. frames, **rasters** and **stars** objects that the **starve** package uses. Using the **tmap** package for exploratory purposes can take just a few lines of code which we show below to create the maps of the original count data and the predictions. The **tmap** package also gives the user the option to customize many aspects of the map to make higher quality and more visually appealing maps, if desired.

```
library(tmap)
missouri_tm<- tm_shape(missouri, is.master =TRUE) + tm_borders()
count_tm<- tm_shape(bird_survey) +
  tm_dots(col ="cnt", title ="Count", size =0.7) +
  tm_facets(by ="year", free.coords =FALSE)
intensity_tm<- tm_shape(bird_stars["response"]) + tm_raster(title
```

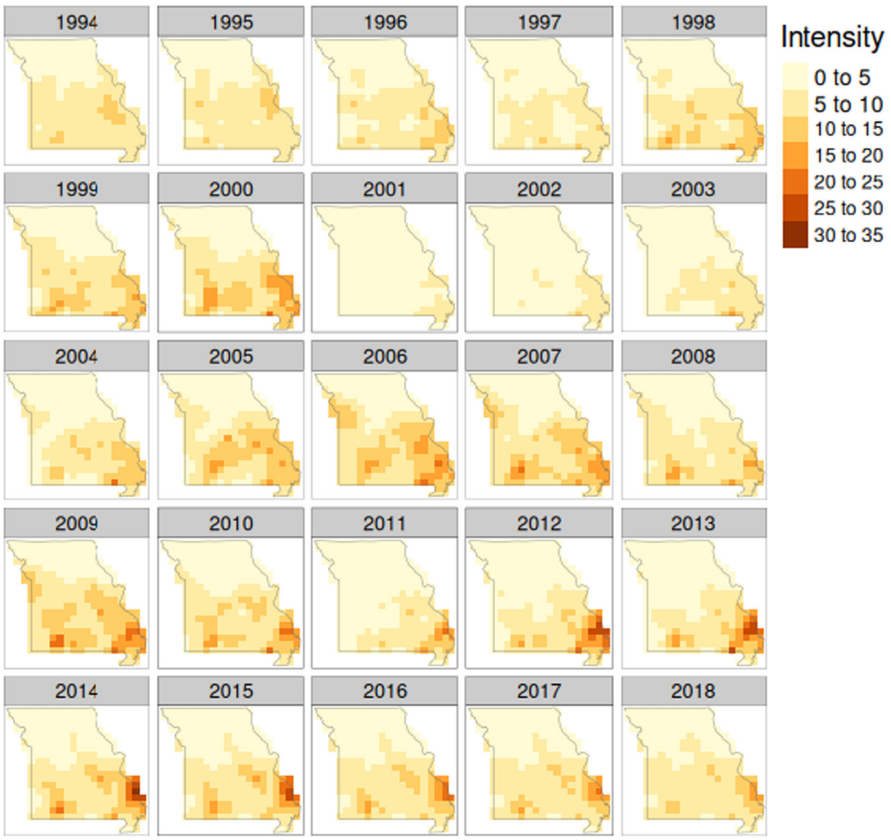

FIGURE 5 Model-based predictions of mean intensity of Carolina wren counts during the survey years (1994–2014) and forecasted into the future (2015–2018). The standard errors (Figure 6) for these predictions give important context regarding the certainty of the predictions, and any insights gained from the predictions need to take into account the uncertainty around the point predictions. The raster predictions should be roughly proportional to the local population abundance and thus give a clear depiction of the spatial distribution exhibited by the wren population. This visualization of the data aided by the model is less cluttered than the map of the observed counts themselves (Figure 4) and shifts in the wren distribution can be more quickly identified. In addition, the predictions provided by the model show more subtle shifts than can be easily seen in the observed data. For example, the emergence of the population centre in the southwest corner of the state from 1994 to 2000 and the range expansion when comparing the peak population sizes in 2000 to 2009 are immediately obvious from the raster predictions but not clear from the map of observed counts.

```
="Intensity")
stderr_tm<- tm_shape(bird_stars["response_se"]) + tm_raster(title
="Std. Error")

legend_tm<- tm_layout(
  legend.outside.size =0.15,
  panel.label.size =1.4
)

count_map<- count_tm + missouri_tm + legend_tm
intensity_map<- intensity_tm + missouri_tm + legend_tm
stderr_map<- stderr_tm + missouri_tm + legend_tm
```

The maps of predicted intensity and the predictions standard errors are shown in Figures 5 and 6 respectively. The spatial distribution of the Carolina wren throughout the years covered by the survey is evident in these predictions. The wren is limited to the southern half of the state with a distinct population centre in the Ozarks in the southwest corner of the state and another near the Mississippi River in the southeast corner of the state. Maps of the predictions are a valuable visualization tool either for exploratory data analysis or for helping to inform management or conservation decisions where scientists need to communicate clearly with nonscientist decision makers.

4 | DISCUSSION

We end with a discussion of some of the current limitations of the **starve** package, a comparison with the **INLA** package, and directions for future research and extensions that we plan on adding to the model and package.

4.1 | Limitations

Currently the main limitation of the **starve** package is that it does not have some of the features and modelling options that older

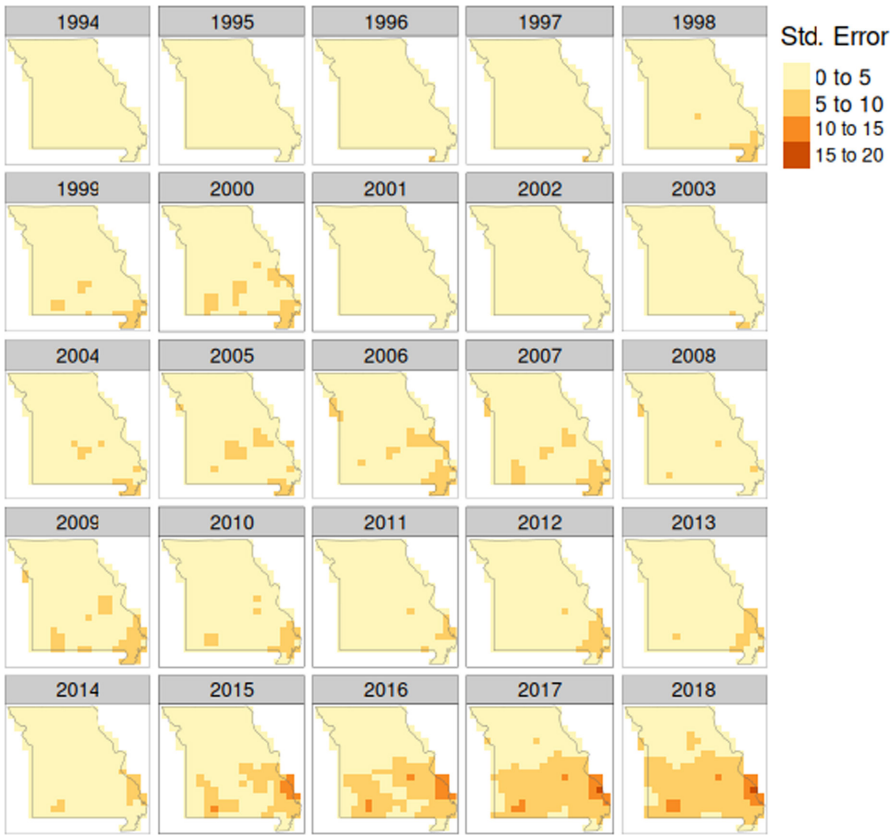

FIGURE 6 Model-based standard errors for predicted mean intensity of Carolina wren counts during the survey years (1994–2014) and forecasted into the future (2015–2018). These standard errors represent the uncertainty inherent in extrapolating trends in the data to predict at unobserved locations and times. Since the survey locations provide good spatial coverage of the state in all years (see Figure 4), the prediction standard errors are low in all areas throughout the duration of the survey. There are some years that have more missing data than other years which causes the uncertainty at the missing locations to be higher than normal. The uncertainty in the forecast years suggest that model predictions are unlikely to be accurate more than 1 or 2 years past the end of the survey, although the model is confident that unless some unforeseen shift in the underlying environment occurs the wren population will still be located in the southern half of the state.

and more mature packages have. For example (Bakka et al., 2019) added a barrier model to the **INLA** package which can account for obstacles in the landscape that prevent animals from moving across the barrier. The **VAST** package can analyse compositional data which can arise in fisheries surveys when interested in the abundance of fish in each age class, and is a more complicated type of response variable than counts or weights. While these features and others like them fit into the theoretical framework of the model, incorporating them into the package while adhering to the design philosophy of being simple and easy-to-use can be challenging.

Other limitations of the **starve** package are the trade-offs made in order to make the interface as simple as it is. Most of the details of fitting the model are completely hidden from the user so if things go wrong in model fitting—false convergence, nonsensical parameter estimates, etc—it can be difficult for the user to experiment and diagnose what is causing the problem. The options to change starting values for the parameter estimates, hold parameters fixed at certain values, and to easily change the spatial covariance function, response distribution, etc. helps mitigate this issue by giving the user access to the typical first steps in diagnosing problems in the model fitting procedure. That the details of model fitting are hidden from the user also means that it is hard if not impossible to tweak the optimization procedure. For example the package does not support (nor has plans to support) Bayesian estimation of parameters and random effects, and is limited to maximum likelihood inference.

It can be difficult to customize the model to specific datasets if the model implemented in the package is not adequate for the dataset. Although the **starve** package is open source and was designed from the start to make model extensions easy to add, you would still need advanced coding skills to add these extensions yourself. Many of the fisheries-specific equations implemented in **VAST**, for example, are not yet implemented in **starve** which currently limits its use for fisheries science.

4.2 | Comparison with INLA

We expect that most ecologists interested in spatio-temporal models for their survey data have at least thought of using the **INLA** package for their analyses so we briefly compare **starve** to **INLA**. Each package uses a different method for making spatio-temporal analysis computationally feasible. **INLA** works by using a mesh to approximate a Gaussian random field with a Gauss–Markov random field whose likelihood can be efficiently computed by directly modelling the precision matrix (the inverse of the covariance matrix) instead of the covariance matrix. This approximation works because of a connection between the Gaussian random field and the Gauss–Markov random field provided by a particular stochastic partial differential equation (Lindgren et al., 2011). **starve** works by using a directed acyclic graph to replace the single large covariance matrix of a Gaussian random field with a large number of very small covariance matrices. **starve** was also purpose-built for spatio-temporal data, whereas **INLA** was built for purely spatial data with support for spatio-temporal models coming from a more general feature for grouping data. This allows the **starve** package to optimize its computations in the spatio-temporal setting, such as caching covariance matrix computations from one year to the next.

We compared the computational performances of **starve**, **INLA** and a naïve spatio-temporal model that uses the full covariance matrix for a separable spatio-temporal covariance function that represents how a novice user of **TMB** might proceed. The model we created in **INLA** is roughly equivalent to the model implemented in **starve**, using an AR(1) time structure and a Matérn spatial structure, although we must mention that we are inexperienced **INLA** users and there may be ways to tweak the model settings to improve performance. The metrics we use are the amount of time needed for all data preprocessing and model fitting for the Carolina wren dataset, the time needed to use the fitted model to predict 2 years of a 20×20 raster, and an estimate of the total amount of RAM needed by **R** to use these packages in this analysis. Results are given in Table 2 with **starve** (n = 10 neighbours) needing about 25% of the time that **INLA** (parallel) needed to preprocess data and fit the model, the same amount of time that **INLA** needed to create predictions, and roughly the same amount of RAM than **INLA** needed. The naïve model took 29 times the amount of time that **starve** did to fit the model, and the memory usage for this naïve model meant that predictions for more than a handful of locations caused **R** to crash.

In addition to comparing favourably to **INLA** in terms of computational performance the workflow for **starve** is more straightforward than for **INLA**, and the amount of specialist knowledge needed to use **starve** is less than that needed for **INLA**. An excerpt of the code used to create the **INLA** model just mentioned is given in Figure 7, which shows the relative complexity of the **INLA** syntax compared to the **starve** syntax shown in Section 3. In addition, **INLA** relies on the user to define sensible prior distributions and to create and choose a good mesh, which requires an appreciation of the approximation method used to make **INLA** efficient. The **starve** package uses sensible default behaviours so that the user does not need to know the details of the approximation method, but can still change the behaviour of the model if desired. However, **INLA** is a much more general-purpose package than **starve**. If taken the time to learn, **INLA** can fit much more complex and tailored models than **starve**.

TABLE 2 Computation time and memory usage when fitting the Carolina wren dataset and creating predictions for 2 years on a 20×20 raster, for implementations of roughly equivalent models using different packages and package settings. The fitting time covers all data preprocessing steps, including the mesh construction (**INLA**) and graph construction (**starve**), and parameter estimation, and similarly for the prediction time. The memory usage refers to the peak RAM usage at any point when running the **R** script for model fitting and predictions using that particular model. The naïve spatial model uses a Gaussian process likelihood with a 783×783 covariance matrix using a separable spatio-temporal covariance function, and is implemented in **TMB** as an example of what a novice user might create themselves. Predictions using the naïve model quickly ran out of memory, crashing a laptop with 16GB of RAM when trying to predict anything more than 1 year of a 6×6 grid without computing standard errors. The mesh for **INLA** used 113 nodes, and the graph nodes for **starve** used the 68 data locations. The **starve** package outperformed the **INLA** package when fitting a model with the default settings for the **starve** package (n = 10) taking roughly 25% of the time that the default settings for the **INLA** package (parallel, 8 cores) took. Both packages performed roughly the same for prediction time and memory usage. The computation time and memory usage for both packages are dependent on the size of the mesh (**INLA**) or graph (**starve**, either through more nodes or more neighbours per node)

	Fitting time (s)	Prediction time (s)	Memory usage (GB)
Naïve spatio-temporal	131.1	—	—
INLA (nonparallel)	26.1	7.0	0.18
INLA (parallel)	19.7	3.0	0.66
starve (n = 30)	11.2	17.2	2.7
starve (n = 10)	4.5	3.3	0.55
starve (n = 3)	3.8	1.7	0.24

4.3 | Future directions

Many surveys collect data on more than one variable: for example they could collect data for multiple species or they might divide the data collected for a single species by their age classes. Adding support for more than one response variable and accounting for dependence between these variables is an important next step in the development of the model and package. We are also planning on adapting a version of the package to fisheries data which would include adding a number of smaller extensions to the model, such as adding a catch equation and a cohort ageing equation for age-structured models. This would make

```
stack<- inla.stack(
  data = list(cnt = bird_survey$cnt),
  A = list(1, A_matrix),
  effects = list(
    Intercept = rep(1, nrow(bird_survey)),
    index = w
  )
)
fit<- inla(
  cnt ~ -1 + Intercept +
    f(w, model = spde, group = w.group,
      control.group = list(model = "ar1", hyper = ar1.spec)),
  family = "poisson",
  data = inla.stack.data(stack),
  control.predictor = list(A = inla.stack.A(stack)),
  control.compute = list(config = TRUE)
)
```

FIGURE 7 Part of the code used to fit a spatio-temporal model equivalent to the **starve** model using **INLA**. The **INLA** package syntax requires the user to call multiple functions and create multiple objects that are pieced together to create a model from scratch. In addition to the functions shown here an **INLA** user must also create a mesh of the study area, create a random effects index and define prior distributions for all model parameters. The functions to do these tasks do not have default behaviours so the user must explicitly define a large number of model settings. Because of the syntax and lack of default behaviour, users of the **INLA** package also need to be familiar with particular sparse method used to make **INLA** computationally efficient.

the **starve** package more directly comparable with **VAST**, although the two packages would use different modelling philosophies and it does not hurt to have more modelling options available to the research community. Other changes to the package that are less immediate on our list of future development include adding options for more complex temporal structures in addition to the current AR(1) structure, adding options for spatial covariance functions besides Matérn functions, and adding support for a barrier model which could be accomplished through clever creation of the nearest-neighbour graph for the nearest-neighbour Gaussian process. In addition to extending the model, more research needs to be done in creating a formal set of procedures for model validation, and more research needs to be done to understand the trade-offs for different choices of nearest-neighbour graphs.

AUTHOR CONTRIBUTIONS

Ethan Lawler designed the model and package with input from Chris Field and Joanna Mills Flemming. All authors contributed to writing the paper and approved it for publication.

ACKNOWLEDGEMENTS

This research was funded by a Canadian Statistical Sciences Institute Collaborative Research Team Project and the Ocean Frontier Institute. Ethan Lawler was also supported by a Vanier Canada Graduate Scholarship and Killam Predoctoral Scholarship. This article and the **starve** package owe a great deal of gratitude to two anonymous reviewers and the editor.

CONFLICT OF INTEREST

The authors have no conflict of interest to disclose.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14053.

DATA AVAILABILITY STATEMENT

The Carolina wren dataset used in this paper was taken from the package **STRbook**, which in turn was modified from (Pardieck et al., 2017). We also include the dataset as part of the **starve** package, which is available at the first author's GitHub page (https://github.com/lawlerem/starve). The specific version of the package used for this manuscript is archived at https://doi.org/10.5281/zenodo.7468568 (Lawler, 2022a). All additional code used to create the analyses in this article is also available on Github (https://github.com/lawlerem/MEE-starve-code) and archived at https://doi.org/10.5281/zenodo.7468536 (Lawler, 2022b).

ORCID

Ethan Lawler https://orcid.org/0000-0002-6225-3009

REFERENCES

Bakka, H., Vanhatalo, J., Illian, J. B., Simpson, D., & Rue, H. (2019). Non-stationary gaussian models with physical barriers. *Spatial Statistics*, *29*, 268–288. https://doi.org/10.1016/j.spasta.2019.01.002

Berger, A. M., Goethel, D. R., Lynch, P. D., Quinn, T. J., Mormede, S., McKenzie, J., & Dunn, A. (2017). Space oddity: The mission for spatial integration. *Canadian Journal of Fisheries and Aquatic Sciences*, *74*, 1698–1716.

Bivand, R., & Nowosad, J. (2022). *CRAN task view: Analysis of spatial data*. https://cran.r-project.org/web/views/Spatial.html

Cosandey-Godin, A., Krainski, E. T., Worm, B., & Mills Flemming, J. (2014). Applying Bayesian spatiotemporal models to fisheries bycatch in the Canadian Arctic. *Canadian Journal of Fisheries and Aquatic Sciences*, *72*(2), 186–197.

Datta, A., Banerjee, S., Finley, A. O., & Gelfand, A. E. (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, *111*(514), 800–812. https://doi.org/10.1080/01621459.2015.1044091

Eynon, B. P., & Switzer, P. (1983). The variability of rainfall acidity. *The Canadian Journal of Statistics*, *11*(1), 11–23. https://doi.org/10.2307/3314707

Giorgi, E., Diggle, P. J., Snow, R. W., & Noor, A. M. (2018). Geostatistical methods for disease mapping and visualisation using data from

spatio-temporally referenced prevalence surveys. *International Statistical Review*, *86*(3), 571–597. https://doi.org/10.1111/insr.12268

Goodall, C., & Mardia, K. V. (1994). Challenges in multivariate spatio-temporal modeling. *Proceedings of XVIIth International Biometrics Conference*, *1*, 1–17.

Hartig, F. (2020). *Residual diagnostics for hierarchical (multi-level/mixed) regression models*. https://CRAN.R-project.org/package=DHARMa

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., & Zammit-Mangion, A. (2018). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological, and Environmental Statistics*, *1*, 398–425. https://doi.org/10.1007/s13253-018-00348-w

Jubinville, I., Lawler, E., Tattrie, S., Shackell, N. L., Mills Flemming, J., & Worm, B. (2021). Distributions of threatened skates and commercial fisheries inform conservation hotspots. *Marine Ecology Progress Series*, *679*, 1–18.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H. J., & Bell, B. M. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, *70*(5), 1–21.

Lawler, E. (2022a). *lawlerem/starve* (Version MEE_0.18.3) [Computer software]. https://doi.org/10.5281/zenodo.7468568

Lawler, E. (2022b). *lawlerem/starve_MEE_reproduce* (Version 1.0.0) [Computer software]. https://doi.org/10.5281/zenodo.7468536

Lindgren, F., & Rue, H. (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, *63*(19), 1–25. https://doi.org/10.18637/jss.v063.i19

Lindgren, F., Rue, H., & Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society B (Statistical Methodology)*, *73*(4), 423–498. https://doi.org/10.1111/j.1467-9868.2011.00777.x

Lindström, J., Szpiro, A., Sampson, P. D., Bergen, S., & Sheppard, L. (2013). *SpatioTemporal: An r package for spatio-temporal modelling of air pollution*. https://cran.r-project.org/package=SpatioTemporal

Nychka, D., Hammerling, D., Sain, S., & Lenssen, N. (2016). *LatticeKrig: Multiresolution kriging based on Markov random fields*. University Corporation for Atmospheric Research. https://doi.org/10.5065/D6HD7T1R

Pardieck, K. L., Ziolkowski, D. J., Jr., Lutmerding, M., Campbell, K., & Hudson, M. A. R. (2017). *North American breeding bird survey dataset 1966–2016, version 2016.0*. U.S. Geological Survey, Patuxent Wildlife Research Center. https://doi.org/10.5066/F7W0944J

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*(1), 439–446. https://doi.org/10.32614/RJ-2018-009

Pebesma, E. (2022). *Stars: Spatiotemporal arrays, raster and vector data cubes*. https://CRAN.R-project.org/package=stars

Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., & Aigrain, S. (2013). Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*(1984), 20110550. https://doi.org/10.1098/rsta.2011.0550

Robinson, N. M., Nelson, W. A., Costello, M. J., Sutherland, J. E., & Lundquist, C. J. (2017). A systematic review of marine-based species distribution models (SDMs) with recommendations for best practice. *Frontiers in Marine Science*, *4*. https://doi.org/10.3389/fmars.2017.00421

Sellers, K. F., Borle, S., & Shmueli, G. (2012). The COM-poisson model for count data: A survey of methods and applications. *Applied Stochastic Models in Business and Industry*, *28*(2), 104–116. https://doi.org/10.1002/asmb.918

South, A. (2017). *Rnaturalearth: World map data from natural earth*. https://CRAN.R-project.org/package=rnaturalearth

Tennekes, M. (2018). tmap: Thematic maps in R. *Journal of Statistical Software*, *84*(6), 1–39. https://doi.org/10.18637/jss.v084.i06

Thorson, J. T., & Barnett, L. A. K. (2017). Comparing estimates of abundance trends and distribution shifts using single- and multispecies models of fishes and biogenic habitat. *ICES Journal of Marine Science*, *74*(5), 1311–1321. https://doi.org/10.1093/icesjms/fsw193

Tikhonov, G., Opedal, Ø. H., Abrego, N., Lehikoinen, A., De Jonge, M. M. J., Oksanen, J., & Ovaskainen, O. (2020). Joint species distribution modelling with the R package Hmsc. *Methods in Ecology and Evolution*, *11*(3), 442–447. https://doi.org/10.1111/2041-210X.13345

Tobler, W. R. (1970). A computer movie simulating urban growth in the Detroit region. *Economic Geography*, *46*, 234–240.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Appendix S1

How to cite this article: Lawler, E., Field, C., & Mills Flemming, J. (2023). starve: An R package for spatio-temporal analysis of research survey data using nearest-neighbour Gaussian processes. *Methods in Ecology and Evolution*, *14*, 817–830. https://doi.org/10.1111/2041-210X.14053